

```
@InProceedings{Beel2025,  
author      = {Beel, Joeran and Gipp, Bela and Jannach, Dietmar and Said, Alan and Wegmeth, Lukas and Vente, Tobias},  
booktitle   = {47th European Conference on Information Retrieval (ECIR)},  
title      = {Checky, the Paper-Submission Checklist Generator for Authors, Reviewers and LLMs},  
year       = {2025},  
}
```

Checky, the Paper-Submission Checklist Generator for Authors, Reviewers and LLMs

Joeran Beel¹[0000-0002-4537-5573], Bela Gipp²[0000-0001-6522-3019], Dietmar
Jannach³[0000-0002-4698-8507], Alan Said⁴[0000-0002-2929-0529], Lukas
Wegmeth¹[0000-0001-8848-9434], and Tobias Vente¹[0009-0003-8881-2379]

¹ University of Siegen & Recommender-Systems.com, Germany

FirstName.LastName@uni-siegen.de

² University of Göttingen, Germany

³ University of Klagenfurt, Austria

⁴ University of Gothenburg, Sweden

Abstract. Submission checklists have become increasingly prevalent for paper submissions to conferences and journals in machine learning research. These checklists typically focus on the rigor of evaluations. They communicate to authors what the venue expects and aid reviewers in assessing the rigor of the research. The recommender-system community has made its first attempts to adopt such lists, but not as widely and comprehensively as the machine learning community. Therefore, we introduce *Checky*, a paper-submission checklist generator, archive, and recommender system. Checky is web-based, open source, and features an archive of checklists and a checklist generator with a recommender system. We hope that Checky *a)* stimulates a discussion within the recommender systems community about whether and how submission checklists should be implemented and *b)* facilitates the creation and management of checklists. Checky is available at <https://checky.recommender-systems.com>.

Keywords: Checklists · Evaluation · Reviews.

1 Introduction

In the machine learning community, submission checklists have become increasingly prevalent in the submission process to academic conferences and journals. For example, the Conference on Neural Information Processing Systems (NeurIPS) was an early adopter of submission checklists [8]. NeurIPS requires authors to complete a checklist and submit it with the manuscript. Submission checklists typically comprise around a dozen questions, often relating to the rigor of the empirical evaluation. For instance, questions may ask whether the source code and data are publicly available, whether hyper-parameters were tuned, and if so, which strategy was used and how much effort was invested. Authors answer questions typically with “Yes”, “No”, or “N/A” and provide a brief justification.

Checklists are intended to help reviewers during the review process but also communicate to authors the expectations on submission to the specific venue.

At NeurIPS, submissions without a completed checklist are desk rejected. Many other machine learning conferences have adopted similar practices, including the International Conference on Machine Learning (ICML) and the International Conference on Automated Machine Learning (AutoML). NeurIPS has recently started a trial exploring how large language models can help authors and reviewers in the review process [6]. In the trial, NeurIPS enables authors to use Large Language Models (LLMs) to check if answers in the checklists align well with the manuscript. Similarly, for ICLR 2025, reviewers are supported by LLMs [10]. Projects like Sakana’s *AI Scientist* even promise to fully automate the review process with “near-human accuracy” [9], though our analysis [3] does not (yet) confirm this.

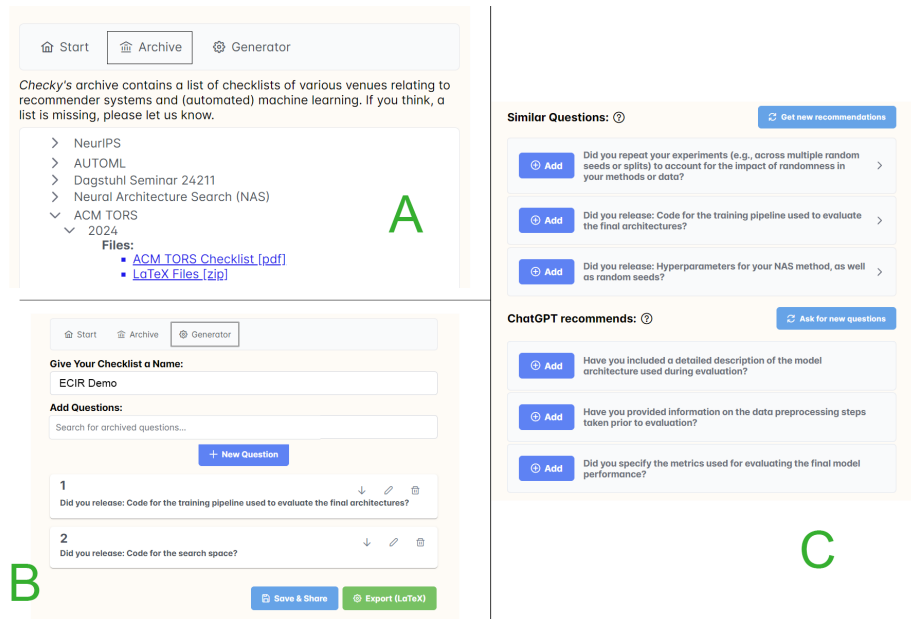


Fig. 1. Screenshot of Checky’s User Interface

For the recommender system community, *Konstan, Ekstrand et al.* argued already in the early 2010s that submission checklists might benefit the community [4,5]. However, only recently did the recommender systems community start its first attempts. For example, the ACM Conference on Recommender Systems introduced a short checklist that authors had to fill out when submitting their manuscripts via EasyChair. The checklists asked, e.g., if the authors had used specific software libraries and, if yes, which ones. The ACM Transactions on Recommender Systems journal (ACM TORS) has encouraged authors since the summer of 2024 to submit a comprehensive checklist. The checklist is based on the template of a recent Dagstuhl Seminar [2] and focuses on the rigor of the

evaluation. The submission of the checklist is, however, optional for authors. Recently, we also advocated a more evidence-based approach to recommender-system evaluation and best-practices [1].

In this work, we introduce *Checky*. Checky is web-based, open source⁵ and freely available at <https://checky.recommender-systems.com>. The purpose of Checky is twofold. First, Checky facilitates the creation and management of checklists with a checklist archive and generator including a recommender system for checklist items. Second, with Checky, we want to raise awareness of the value of submission checklists and initiate discussions about if and how the information retrieval and recommender system community should utilize such lists, and what role LLMs may play.

2 Checky’s Features

2.1 Archive of Existing Submission-Checklists

Checky’s checklist archive (Figure 1 A) features a list of existing checklists from conferences and journals in (automated) machine learning and recommender systems. A PDF and/or LaTeX file are provided for each venue and year. The archive may inspire conference chairs and journal editors who do not yet know what a submission checklist for their journal or conference could look like. Checky additionally provides prompts for LLMs as a unique feature. Currently, we have created these prompts ourselves. However, we imagine that conference chairs and journal editors will provide prompts to ease the work of their authors and reviewers in the future. These prompts may help authors fill out the checklists and support reviewers to verify that the authors correctly filled out the checklists. For instance, an author prompt might read as follows:

My manuscript (PDF) and the submission checklist for the ACM TORS journal are attached. I have filled out the checklist to the best of my knowledge. Please verify the filled checklist against the manuscript and point out any mistakes you discover and everything that might be unclear to a potential reviewer. Make specific suggestions for improvements before I submit my manuscript and checklist to ACM TORS.

Similarly, a prompt for a reviewer could read:

I am a reviewer for the ACM TORS journal. Attached is a manuscript that I need to review, along with the authors’ submission checklist. Please verify that the answers in the submission checklist align with the manuscript’s content. Identify and point out any discrepancies.

2.2 Checklist Generator

Checky’s checklist generator allows for creating new checklists (Figure 1 B). It features a search for the questions in Checky’s checklist archive. Users may

⁵ <https://code.isg.beel.org/Checky>

import these questions into their checklist, modify them, or create new questions. For each question, users specify the answer type, i.e., a free text answer, a free text answer plus justification, or a no-text answer field.

Currently, checklists are exported in L^AT_EX format. Checky also allows for the saving and sharing of checklists online. Each checklist is given a unique name and URL, and collaborators can edit and save the list through that URL. In the future, we plan to implement a more comprehensive user and checklist management system and support additional checklist export formats.

2.3 Recommender System

Checky features a recommender system as part of the checklist generator (Figure 1 C). The recommender system suggests questions for journal editors or conference chairs to add to their new submission checklists. Checky’s recommender system applies two recommendation approaches. First, it supports content-based recommendations. Based on the term frequencies of the items in the currently created checklist, Checky recommends questions from the checklists in the archive. Recommendations are generated on Checky’s server based on Cosine similarity. The second approach utilizes ChatGPT’s API. Checky sends an API request to ChatGPT that includes the questions of the currently created checklist. Subsequently, Checky asks ChatGPT for a set of additional questions.

3 Outlook & The Role of Large Language Models

We hope that Checky will support the information retrieval and recommender systems community in creating and establishing paper-submission checklists for journals and conferences. In the long run, we envision that conference chairs and journal editors will register at Checky, access checklists from previous years, modify them, and manage checklists for the current submissions. Checky could also be integrated with conference management systems like EasyChair.

Also, we hope that Checky contributes to a discussion about the role of Large Language Models in supporting the submission and review of manuscripts. As of now, checky is just a simple proof of concept, but looking at current work [9,7,3] and thinking five or even 25 years ahead, we consider it likely that machine learning and LLMs will play a significant role in writing, reviewing, and publishing research papers. Therefore, we advocate that the IR and RecSys communities should proactively explore the potential of LLMs and conduct, for instance, trials similar to those of NeurIPS and ICLR [6,10].

4 Acknowledgments

We are grateful to Moritz Baumgart who programmed Checky’s first prototype.

References

1. Beel, J.: A call for evidence-based best-practices for recommender systems evaluations. In: Bauer, C., Said, A., Zangerle, E. (eds.) Report from Dagstuhl Seminar 24211: Evaluation Perspectives of Recommender Systems: Driving Research and Education (2024). <https://doi.org/10.31219/osf.io/djuac>, https://isg.beel.org/pubs/2024_Call_for_Evidence_Based_RecSys_Evaluation__Pre_Print_.pdf
2. Beel, J., Jannach, D., Said, A., Shani, G., Vente, T., Wegmeth, L.: Best-practices for offline evaluations of recommender systems. In: Bauer, C., Said, A., Zangerle, E. (eds.) Report from Dagstuhl Seminar 24211 – Evaluation Perspectives of Recommender Systems: Driving Research and Education (2024)
3. Beel, J., Kan, M.Y., Baumgart, M.: Sakana’s AI Scientist: Bold Promises, Mixed Results – A Critical Replication Study. Under Review / Work-in-Progress (2025)
4. Ekstrand, M.D., Ludwig, M., Konstan, J.A., Riedl, J.T.: Rethinking the recommender research ecosystem: reproducibility, openness, and LensKit. In: Proceedings of the Fifth ACM Conference on Recommender Systems. p. 133–140. RecSys ’11 (2011). <https://doi.org/10.1145/2043932.2043958>
5. Konstan, J.A., Adomavicius, G.: Toward identification and adoption of best practices in algorithmic recommender systems research. In: Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation. p. 23–28. RepSys ’13 (2013). <https://doi.org/10.1145/2532508.2532513>
6. Lu, A.: Soliciting Participants for the NeurIPS 2024 Checklist Assistant Study (May 2024), <https://blog.neurips.cc/2024/05/07/soliciting-participants-for-the-neurips-2024-checklist-assistant-study/>
7. Lu, C., Lu, C., Lange, R.T., Foerster, J., Clune, J., Ha, D.: The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery (2024), <https://arxiv.org/abs/2408.06292>
8. NeurIPS: NeurIPS 2022 Paper Checklist Guidelines (2022), <https://neurips.cc/Conferences/2022/PaperInformation/PaperChecklist>
9. Sakana: The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery (Blog) <https://sakana.ai/ai-scientist/>
10. Vondrick, C.: Assisting ICLR 2025 reviewers with feedback. ICLR Blog (2024), <https://blog.iclr.cc/2024/10/09/iclr2025-assisting-reviewers/>